

INTERRUPT CONTROLLER FOR DSP-BASED CONTROL OF MULTI-RAIL DC-DC CONVERTERS WITH NON-INTEGERSWITCHING FREQUENCY RATIO

James Mooney, Simon Effler, Mark Halton, Abdulhussain E. Mahdi

Department of Electronic & Computer Engineering
University of Limerick
Limerick, Ireland
james.mooney@ul.ie

ABSTRACT

This paper proposes a hardware modification to a standard Digital Signal Processor (DSP) that enables it to digitally control multiple DC-DC converters with non-integer switching frequency ratios. The modified DSP overcomes the drawbacks of conventional DSPs, which are attributable to large variations in the delay from when the Analog to Digital Converter (ADC) samples the output voltage to when the duty cycle is updated. By incorporating modified interrupt control logic in the DSP, the effects of a variable delay are minimized by significantly reducing the worst case sampling to duty-cycle-updating delay. Applying this DSP to a multi-rail power supply system provides the designer with the flexibility to choose arbitrary switching frequencies for individual power converters, thereby allowing optimization of the efficiency and performance of the individual converters.

Keywords— interrupt controller; multi-rail DC-DC converters; digital control; digital signal processor.

1. INTRODUCTION

Intermediate bus architectures are common in computing and telecommunications devices today where the output of an isolated DC-DC converter forms an intermediate voltage bus that acts as the input to multiple non-isolated Point-of-Load (POL) converters [1, 2]. The POL converters have different voltage levels as well as being applied to loads with a variety of specifications, for example in Field Programmable Gate Array (FPGA) applications [3, 4]. The use of digital control is beneficial in such systems as a single digital controller can compensate multiple DC-DC converters, as Fig. 1 shows. This is in contrast with an analog system which requires individual compensators for each converter resulting in a potentially more expensive power supply that consumes more board area.

Digital controllers can be classified by the type of digital computational hardware that executes the control law. Two main approaches are possible: the first uses fixed-algorithm dedicated computational hardware for each individual power converter [5], while the second uses a single DSP that can be time-multiplexed among the different converters [6]. DSPs are commonly used as digital controllers in multi-rail power converter systems. They provide flexibility in terms of the software-based control algorithms that can be implemented and they also incorporate communication and monitoring functionality. DSPs are ideally suited to the implementation of

computationally intensive algorithms such as those used in adaptive tuning loops [7-9]. These algorithms would be impractical to implement using dedicated hardware for a multi-rail system because of the excessive area requirements.

A problem with conventional DSPs is that the switching frequencies of the individual converters being controlled are normally restricted to being identical or integer multiples of each other. This restriction is imposed due to variations in the delay between ADC-sampling and Digital Pulse Width Modulator (DPWM) duty-cycle-updating, which are caused by simultaneously occurring interrupts if the switching frequencies have a non-integer ratio. A detailed analysis of this problem is presented in the next section. Constraining the switching frequencies to integer multiples of each other can impact the efficiency or performance of the converters because the designer is forced into selecting non-optimal switching frequencies. If an arbitrary switching frequency can be chosen, the task of meeting the specifications of individual POL converters each with unique efficiency and performance specifications is easier. This paper proposes a hardware solution that minimizes the DSP's variable delay and the associated problematic effects, thereby enabling the benefits of non-integer switching frequency ratios to be obtained in multi-rail applications.

2. INTERRUPT MANAGEMENT FOR MULTIPLE CONTROL LOOPS

2.1 Interrupt-triggered control with integer multiple frequency ratios

Interrupts are used to trigger the execution of control algorithms in DSP-based digital power controllers. The interrupt signal is usually synchronized with the ADC's sampling clock so that control algorithm execution immediately commences when a new ADC sample is available. In general, only one sample is processed per switching cycle. When multiple power converters are being controlled the individual algorithms share the DSP's computation time. This is achieved by interleaving the interrupt signals as illustrated in Fig. 2. The method is only practical when all algorithms are either executed at the same frequency or at different frequencies that are integer multiples of each other. When the execution frequencies have non-integer ratios it is very difficult or even impossible to schedule execution of algorithms so that they do not overlap.

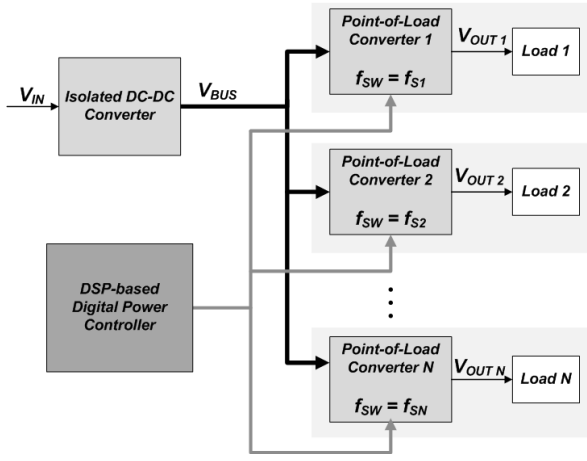


Figure 1. DSP-based digital controller applied to multiple DC-DC converters

2.2 Interrupt-triggered control with non-integer multiple frequency ratios

In a typical power control DSP only one algorithm can be executing at any time, therefore all others must wait until the currently executing algorithm has completed i.e. the control algorithm's execution cannot be interrupted. When multiple interrupt signals occur simultaneously, the algorithms are typically executed according to a pre-defined priority as Fig. 3 illustrates. Thus an additional delay is introduced between ADC-sampling and duty-cycle-updating for the lower priority algorithm. Consequently the delay between ADC-sampling and duty-cycle-updating can vary each time an interrupt is triggered, depending on whether or not multiple interrupts have occurred simultaneously. This variation in delay is undesirable when designing a compensator because a fixed delay is assumed in modeling the closed loop system [10]. Ignoring the variation would cause unpredictable behavior in the power converter and could ultimately lead to failure.

In order to avoid the problems associated with a variable delay, the delay can be fixed at its maximum possible value for each iteration of each algorithm. This is achieved by setting the sampling instant of the ADC at a fixed offset from the beginning of the next switching period. Thus when only one interrupt occurs there will be an 'idle' interval between when the duty cycle is calculated and the beginning of the next switching cycle. Conversely when the maximum number of interrupts occurs simultaneously the duty cycle will be calculated just in time for the beginning of the next switching cycle. However this fixed delay is excessive in that it severely degrades the performance of the voltage regulator due to a much slower response to load transients. Improved performance can be obtained by reducing this delay [11].

2.3 Minimizing delay using pre-calculations

In practice the ADC-sampling to duty-cycle-updating delay in a DSP can be reduced by rearranging the operations in the control algorithm. By executing some 'pre-calculation' operations prior to receiving the ADC sample, fewer instructions need to be executed before updating the DPWM. These pre-calculations are executed in the previous iteration of the algorithm after the duty cycle has been calculated for that iteration. For example in the case of a digital PID compensator,

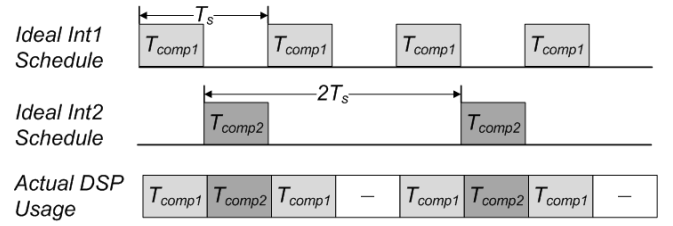


Figure 2. Scheduling for interrupts with integer multiple frequency ratio

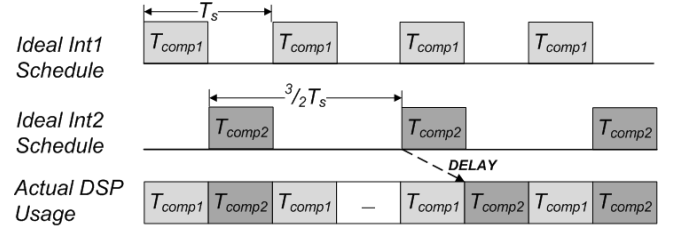


Figure 3. Scheduling for interrupts with non-integer multiple frequency ratio

after the most recent voltage error sample has been obtained it only needs to be multiplied by one coefficient and added to the pre-calculated sum of the other terms in the duty cycle equation, before it can be applied to the DPWM. After this the pre-calculations can be completed in advance of the next iteration. Even when applying this pre-calculation technique using standard DSPs, the delays that occur when multiple interrupt signals coincide are significant. Fig. 4 shows the delays in the situation where three interrupt signals coincide. The maximum ADC-sample to duty-cycle-update delay for a particular control algorithm is described as:

$$T_{DMAX} = T_{ADC} + \sum_{i=0}^{HP} (T_{DCi} + T_{PCi}), \quad (1)$$

where T_{ADC} is the ADC delay, HP is the number of algorithms with higher priority than this algorithm, T_{DC} is the duty cycle calculation time and T_{PC} is the pre-calculation time.

3. MODIFIED INTERRUPT CONTROLLER

In order to avoid the effects of variable DSP delays by fixing the delay at its maximum, a modified interrupt controller for DSPs is proposed that reduces T_{DMAX} to an acceptable value. Fig. 5 illustrates the resulting delays if all duty cycle calculations for coinciding interrupts are executed before any pre-calculations for the next iteration are carried out. By postponing the pre-calculations until duty-cycle-updating has been completed, the total ADC-sample to duty-cycle-update delay, T_{DMAX} as given in (1), is reduced. The value of the reduced delay may be obtained from the following equation:

$$T_{DMAX}^* = T_{ADC} + \sum_{i=0}^{HP} (T_{DCi}). \quad (2)$$

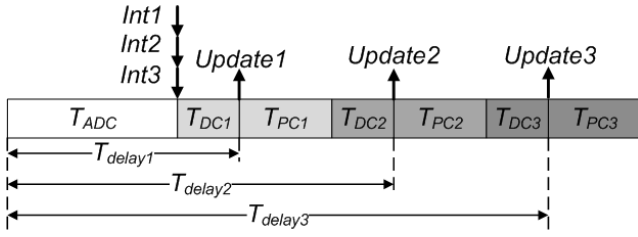


Figure 4. DSP delay with standard interrupt control

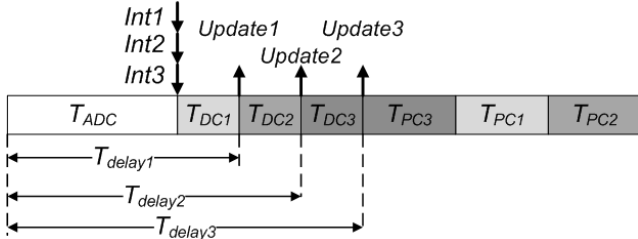


Figure 5. DSP delay with proposed interrupt control

3.1 Modified interrupt controller behavior

The modified interrupt controller achieves the behavior illustrated in Fig. 5 by automatically re-enabling all interrupts after the control algorithm has passed a certain stage of execution. This allows interruption of one control algorithm by another control algorithm, i.e. interrupt nesting, during the pre-calculation stage, after the duty cycle has been calculated and the DPWM has been updated.

Each control algorithm has a different interrupt priority level. When multiple interrupts occur the highest priority algorithm is selected first. At this point all other interrupts are disabled, a dedicated counter is loaded with a pre-configured duty cycle calculation time and counting is enabled. Interrupts are re-enabled after the counter determines that the duty cycle calculation time has elapsed. This duration is configurable for each of the algorithms in order to provide the flexibility to execute a different algorithm for each individual power converter. Before the pre-calculations for the highest priority interrupt can begin, execution is interrupted by the next highest priority algorithm. Again all other interrupts are disabled, the counter is reloaded and counting is enabled. The same applies for the next priority interrupt and so on. After no further interrupts are pending, the DSP continues with the execution of the pre-calculations for each of the interrupted algorithms in preparation for their next iterations.

The substantial benefits of the modified interrupt scheme can be achieved by augmenting a conventional DSP's interrupt controller with minimal additional hardware. The main requirement is a counter to determine when to re-enable interrupts. Some extra registers are also required to store the interrupt return addresses and the duty-cycle calculation times for each algorithm in terms of the number of instructions required.

3.2 Comparison with existing DSPs

The technique described above cannot be directly implemented by existing DSPs that are used in power converter control applications [12-14]. Although the interrupts could be manually re-enabled after the duty cycle has been calculated, this requires an additional instruction at the beginning and end

of each duty cycle calculation, which adds to the ADC-sample to duty-cycle-update delay. By doing this automatically the proposed scheme frees up more execution time in each switching cycle which could be used to execute instructions of a more complex control algorithm or to perform additional monitoring operations. Some DSPs also only facilitate interrupt nesting where higher priority interrupts are allowed to interrupt the currently executing Interrupt Service Routine (ISR), which does not comply with the requirement for each algorithm to be interruptible after a certain number of instructions have been executed. The priority of the interrupts in the proposed method relates to the order in which they will be processed if they occur simultaneously and bears no influence on whether or not they can interrupt other ISRs.

An alternative method that can be implemented using most commercial DSPs is to have separate interrupts for the duty cycle calculation and pre-calculation section of each control algorithm. After the duty cycle calculation has been completed it triggers a lower priority software interrupt for the pre-calculation. This allows other pending duty cycle calculation interrupts to be processed before any pre-calculations take place. The drawback of this method is that a "return from interrupt" instruction must be executed after each duty cycle calculation to exit from the ISR before processing the next pending interrupt. This adds a significant number of additional clock cycles between the duty cycle calculations for each algorithm and thus increases T_{DMAX} . The proposed method avoids this problem by keeping the duty cycle calculation and pre-calculation in the same ISR and only exiting from that routine if another interrupt is pending.

4. MULTI-RAIL DC-DC CONVERTER APPLICATION

In order to evaluate the performance improvement provided by the proposed interrupt controller, it has been compared with the standard interrupt control method, which is illustrated in Fig. 4. The comparison is based on the application of both interrupt control methods to a power converter system consisting of three 12 V - to - 1.5 V buck converters. The control algorithm for Rail 1 has the highest interrupt priority, followed by Rail 2 and then Rail 3. Each calculation cycle is approximately 30 ns, corresponding to a 33 MHz clock frequency.

The proposed interrupt controller was incorporated into a custom dual-datapath Digital Signal Processor design for power control applications [15]. A third order linear compensator was implemented for execution on the dual-datapath DSP to regulate the output voltage of each of the buck converters. The compensator algorithm consisted of six duty-cycle operations and six pre-calculation operations.

Fig. 6 illustrates some of the modified interrupt controller's signals while in operation, where ISR 0 is interrupted by ISR 1 after the duty cycle has been calculated. Execution returns to ISR 0 to complete the pre-calculations for the next iteration after ISR 1 has finished. ISR 2 runs without interruption. Background code is executed in the idle time between the ISRs, which is indicated by ISR 3 in Fig. 6.

Table I summarizes the T_{DMAX} delays for each of the voltage rails for the standard and proposed interrupt methods. It also includes the percentage reduction in delay provided by the proposed interrupt scheme. Although the modified interrupt method does not provide any reduction in T_{DMAX} for the highest

5. CONCLUSION

A drawback of using a standard DSP to control multiple power converters is its limitation in dealing with switching frequencies with non-integer ratios. A modified interrupt controller for Digital Signal Processors has been proposed that performs significantly better in such applications. In comparison with the varying or excessive ADC-sample to duty-cycle-update delay in existing DSPs, the proposed interrupt method yields a constant, reduced and hence more desirable delay. The modified DSP thus provides the designer with flexibility in choosing the switching frequency and also the converters' component values. This means that the efficiency and performance of the individual power converters can be improved.

REFERENCES

- [1] R. Miftakhutdinov, "Power distribution architecture for tele- and data communication system based on new generation intermediate bus converter," in International Telecommunications Energy Conference, 2008, pp. 1-8.
- [2] H. Hong, "Coordination of design issues in the intermediate bus architecture," in Applied Power Electronics Conference, 2005, pp. 169-175 Vol. 1.
- [3] National Semiconductor Design Guide. (2005, March). Power Management Design Guide for Altera FPGAs and CPLDs [Online]. Available: <http://www.national.com/appinfo/power/files/NationalAlteraDesignGuide.pdf>
- [4] Intersil Application Guide. (2004, March). Power Management Application Guide for Xilinx FPGAs [Online]. Available: <http://www.intersil.com/data/ag/ag0001.pdf>
- [5] Texas Instruments Data Sheet. (2008, Nov.). UCD9240 Digital PWM System Controller [Online]. Available: <http://focus.ti.com/lit/ds/symlink/ucd9240.pdf>
- [6] J. Mooney, A. Mahdi, A. Kelly, and K. Rinne, "DSP-based controller for multi-output/multi-phase high switching frequency DC-DC converters," in IEEE Workshop on Control and Modeling for Power Electronics, 2008, pp. 1-6.
- [7] A. Kelly and K. Rinne, "A self-compensating adaptive digital regulator for switching converters based on linear prediction," in IEEE Applied Power Electronics Conference and Exposition, 2006, pp. 712-718.
- [8] J. Morroni, R. Zane, and D. Maksimovic, "Design and implementation of an adaptive tuning system based on desired phase margin for digitally controlled DC-DC Converters," IEEE Trans. on Power Electronics, vol. 24, pp. 559-564, 2009.
- [9] J. Mooney, A. E. Mahdi, A. Kelly, and K. Rinne, "Dual MAC processor for adaptive control of multiple high switching frequency DC-DC converters," in IET Irish Signals and Systems Conference, 2008, pp. 116-121.
- [10] D. Maksimovic and R. Zane, "Small-signal discrete-time modeling of digitally controlled PWM converters," IEEE Transactions on Power Electronics, vol. 22, pp. 2552-2556, 2007.
- [11] L. Corradini, S. Saggini, and P. Mattavelli, "Analysis of a high-bandwidth event-based digital controller for DC-DC converters," in IEEE Power Electronics Specialists Conference, 2008, pp. 4578-4584.
- [12] Texas Instruments Reference Guide. (2009, Jan.). TMS320C28x CPU and Instruction Set Reference Guide [Online]. Available: <http://focus.ti.com/lit/ug/spru430e/spru430e.pdf>
- [13] Analog Devices Reference Guide. (2003, Jan.). ADSP-2199x Mixed Signal DSP Controller Hardware Reference [Online]. Available: http://www.analog.com/static/imported-files/processor_manuals/50171261290224ADSP_2199x_Hardware_Reference.pdf
- [14] Freescale Semiconductor Reference Guide. (2005, Nov.). DSP56800E Reference Manual [Online]. Available: http://cache.freescale.com/files/dsp/doc/ref_manual/DSP56800ERM.pdf
- [15] J. Mooney, M. Halton, and A. E. Mahdi, "Specialized digital signal processor for control of multi-rail/multi-phase high switching frequency power converters," in IEEE Applied Power Electronics Conference, 2010, pp. 2207-2211.

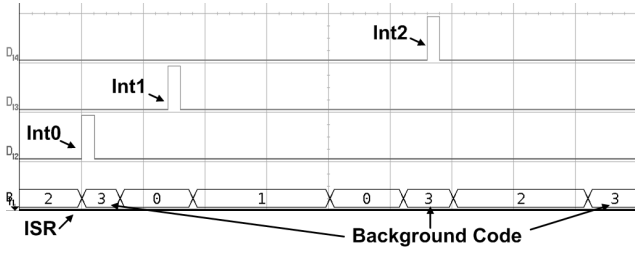


Figure 6. Modified interrupt controller operation, ISR 0 interrupted by ISR 1

TABLE I. MAXIMUM ADC-SAMPLE TO DUTY-CYCLE-UPDATE DELAYS FOR THIRD ORDER COMPENSATOR

Interrupt Method	Rail 1	Rail 2	Rail 3
T_{DMAX} for Standard Interrupt Control	360 ns	720 ns	1080 ns
T_{DMAX}^* for Modified Interrupt Control	360 ns	540 ns	720 ns
Reduction in T_{DMAX} (% Reduction in T_{DMAX})	0 ns (0 %)	180 ns (25 %)	360 ns (33 %)

priority interrupt i.e. Rail 1, it significantly reduces T_{DMAX} for the lower priority interrupts. It should be noted that the delay for Rail 1 corresponds to the delay that would occur for each of the rails in a system with an integer switching frequency ratio.

The resulting difference in performance between the standard and modified interrupt controllers is demonstrated in Fig. 7 and Fig. 8. The restriction of the long T_{DMAX} delay for the standard method results in a slow response to a load current step as illustrated in Fig. 7. The modified interrupt method with the shorter T_{DMAX} delay provides better performance and thus also facilitates the use of a wider bandwidth compensator. This enables a faster response to the same load step to be obtained as Fig. 8 shows.

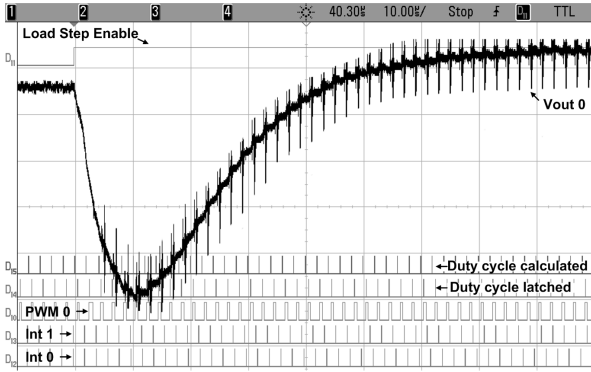


Figure 7. Load transient response for standard interrupt controller

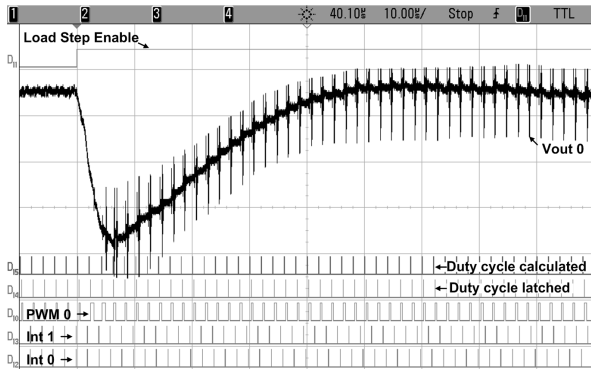


Figure 8. Load transient response for modified interrupt controller with wider bandwidth compensator